Algorithms for Large, Sparse Network Alignment

Mohsen Bayati, David Gleich, Margot Gerritsen, Amin Saberi,
Ying Wang @ Stanford University

and

Jeong Han Kim @ Yonsei University

Our motivation

Library of Congress subject headings

Desserts

URI: <http://id.loc.gov/authorities/sh85037243#concept>

Type: Topical Term

Broader Terms:

- Confectionery

Narrower Terms:

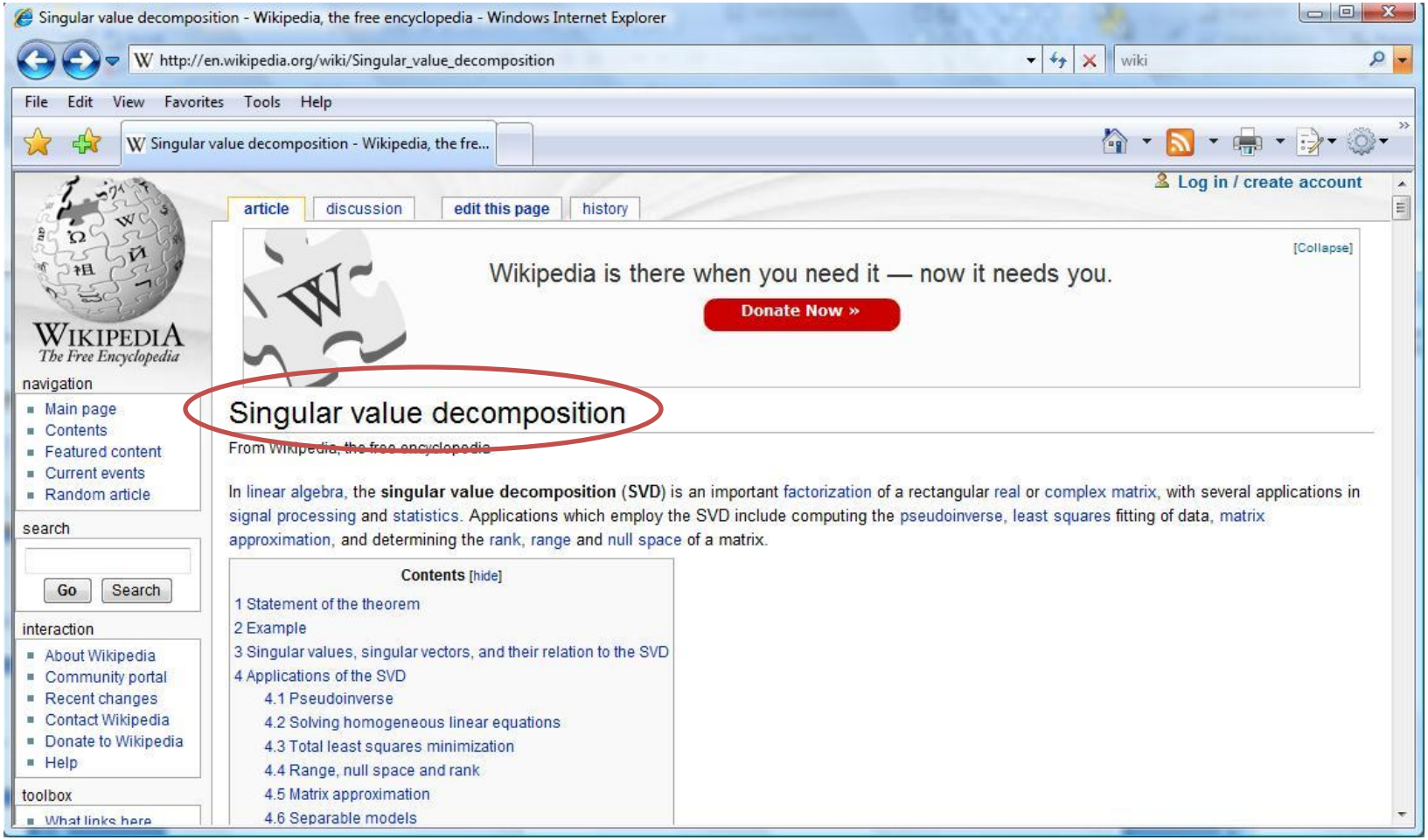
- Ambient desserts
- Banana splits
- Charlottes (Desserts)
- Chocolate desserts
- Frozen desserts
- Ice cream cones
- Mousses
- Puddings
- Refrigerated desserts
- Sundaes
- Whipped toppings

LC Classification: TX773

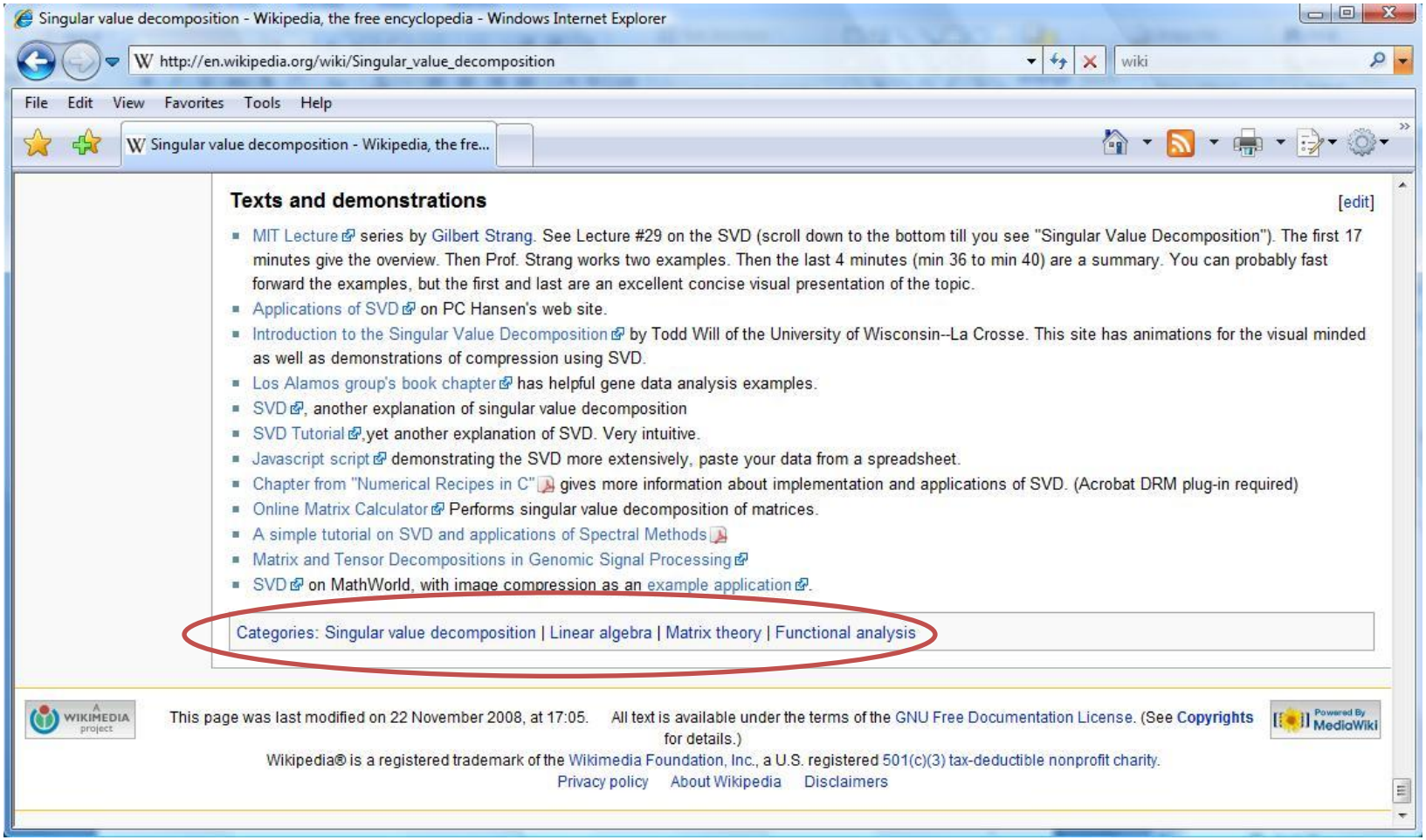
Created: 1986-02-11

Last Modified: 1988-01-15 17:36:44

Wikipedia categories



Wikipedia categories



Wikipedia categories

Category:Desserts

From Wikipedia, the free encyclopedia

*The main article for this category is **dessert**.*

Desserts are sweet foods eaten purely for pleasure, typically at the end of a meal.

Wikimedia Commons has media related to: ***Desserts***

Subcategories

This category has the following 16 subcategories, out of 16 total.

*****

- [+] Desserts by country (20)

B

- [+] Brand name desserts (3)

C

- [+] Cakes (0)
- [+] Chocolate desserts (0)
- [+] Confectionery (9)

C cont.

- [+] Cookies (1)
- [+] Custard desserts (0)

D

- [+] Dessert sauces (0)
- [+] Doughnuts (1)

F

- [+] Frozen desserts (2)

I

- [+] Ice cream (5)

P

- [+] Pastry (4)
- [+] Pies (9)
- [+] Puddings (2)

S

- [+] Sweet breads (2)

μ

- [+] Dessert stubs (1)

Wikipedia vs Library of Congress

Created by many, non-experts, in a distributed way in a few years

Library of Congress

Developed by few, experts, in a centralized way in over a century.

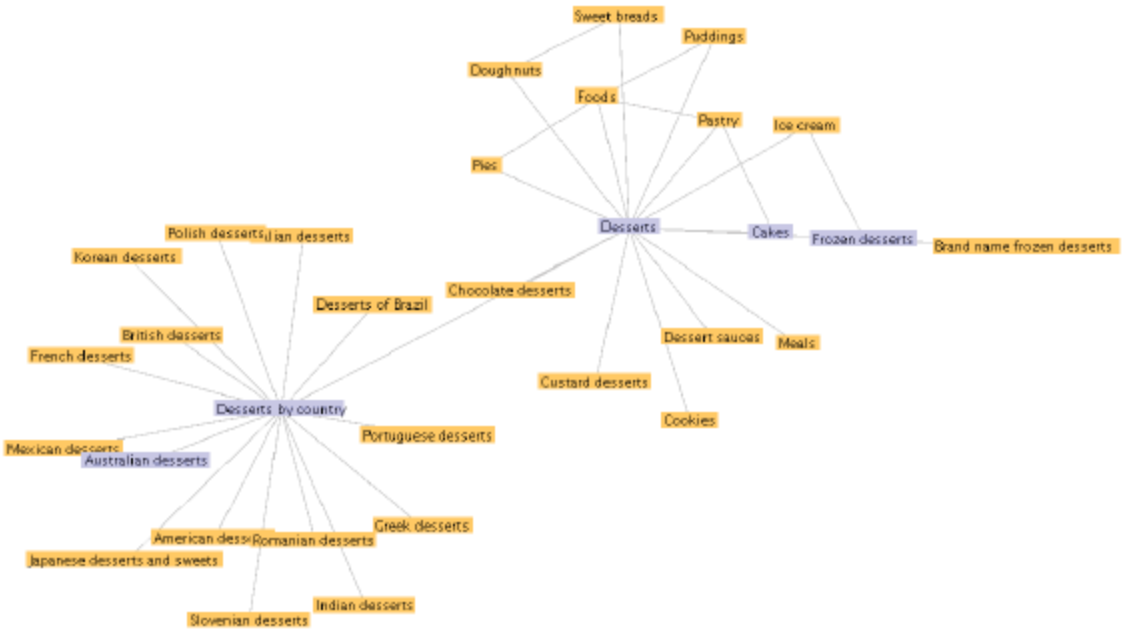
Are they similar ?

Wikipedia vs Library of Congress

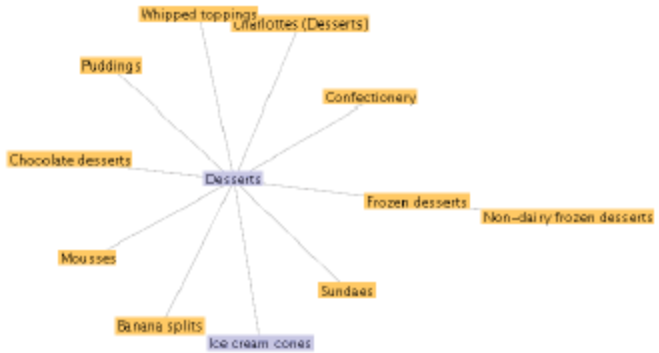
- How similar these two data-sets are ?
- Can we use one data-set to enrich the other ?
- How to spend tax-payer's money more wisely to maintain the Library of Congress ?

Project funded by the Library of Congress.

Are these graphs similar



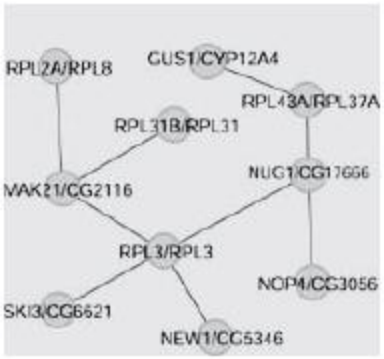
WC



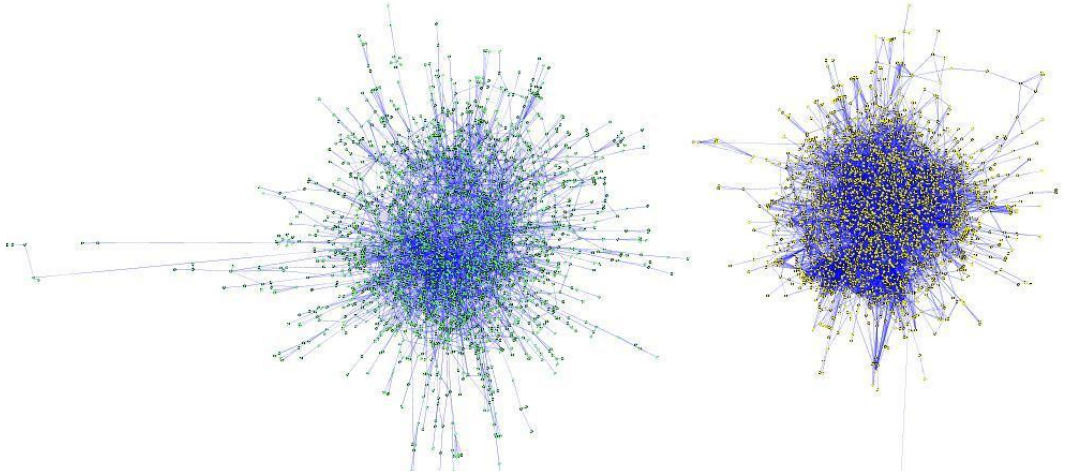
LCSH

Network alignment for Comparing data-sets

- Match cross species vertices (proteins) and edges (protein interactions) → Detect functionally similar proteins.



Berger et al'08, PNAS.

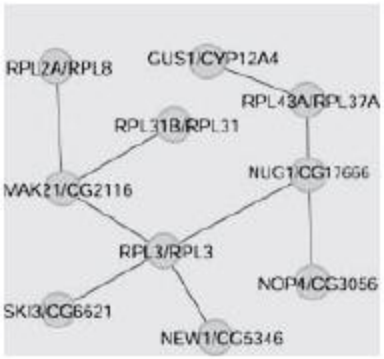


Fly Yeast

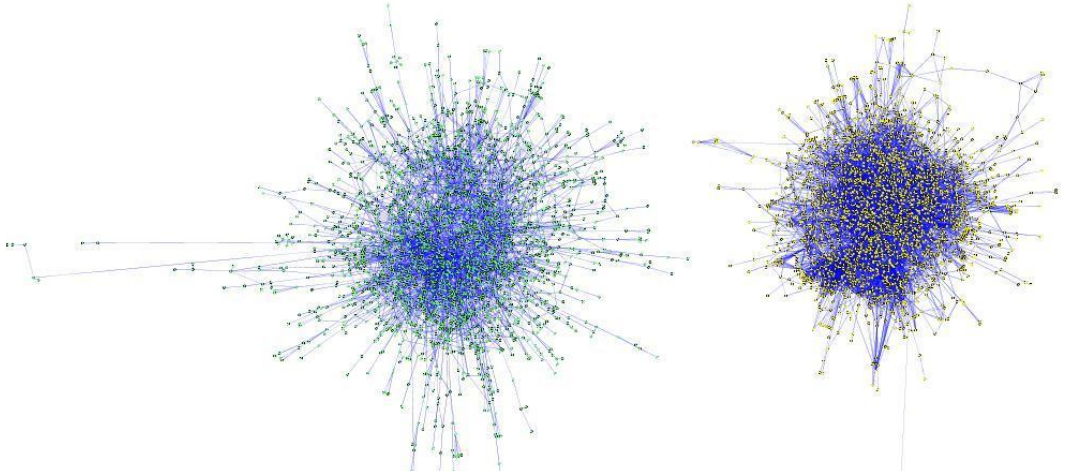
from Drexel University, School of Biomedical Engineering Website

Network alignment for Comparing data-sets

- Find the largest common sub-graph on similar vertices. (Singh-Xu-Berger'07,'08).
- Recently (Klau'09).



Berger et al'08, PNAS.



Fly Yeast

from Drexel University, School of Biomedical Engineering Website

Network alignment for Comparing data-sets

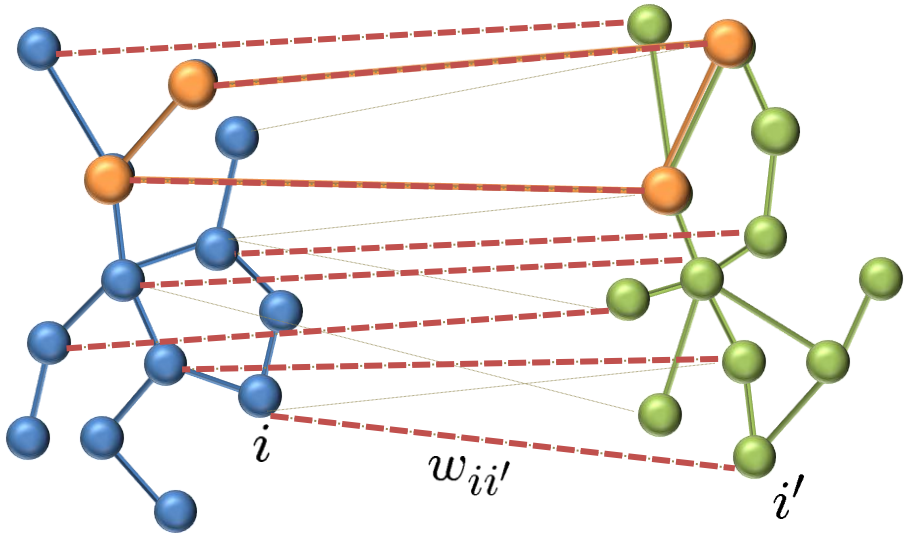
- Database schema matching
  - (Melnik-Garcia Molina-Rahm'02).
- Computer vision: Match a query image to an existing image.
  - (Conte-Foggia'04)
- Ontology matching: Match query image to existing image.
  - (Svab'07).
- Website : Match similar parts of the web-graph.
  - Toyota's USA websites vs Toyota France.
- Social networks: Teenagers have both fake and real identities.
- This talk: Comparing Wikipedia vs Library of Congress.

This talk

- Defining the problem mathematically
- Quick survey of existing approaches
- A message-passing algorithm.
- Experiments
  - Real data
  - Synthetic data
- Rigorous results.

Approach: Align the two databases

205,948 nodes
422,503 links



297,266 nodes
248,232 links

5,233,829 potential matches

Goal: Find an **alignment** that **matches similar titles** and **maximizes** the total number of **overlaps**.

Quadratic program formulation

Formulate the problem as a quadratic program (QP).

Total similarity

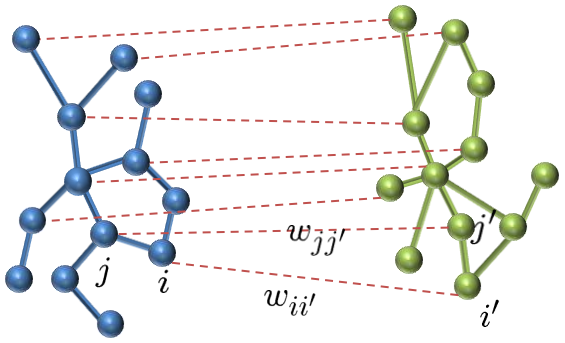
Total overlap

maximize $\alpha \sum_{ii'} x_{ii'} w_{ii'} + \beta x^T S x$

Subject to: $\mathbf{A}\vec{x} \preceq 1$

$x_{ii'} \in \{0, 1\}$

Linear constraints



Maximizing the similarity alone is easy, but the overlap is NP-hard to maximize. There is a reduction from the MAX-CUT problem.

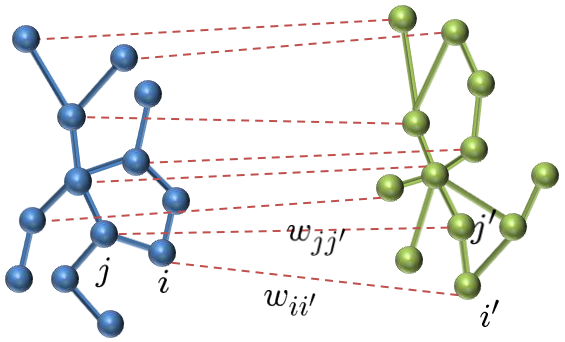
NP-hard to obtain better than 87.8% of the optimum overlap, unless the unique games conjecture is false (Goeman's-Williamson'95).

Quadratic program formulation

maximize $\alpha \sum_{ii'} x_{ii'} w_{ii'} + \beta x^T S x$

Subject to: $\mathbf{A}\vec{x} \preceq \mathbf{1}$

$x_{ii'} \in \{0, 1\}$



Related NP-hard problems:

1) Maximum common sub-graph.

2) Graph isomorphism.

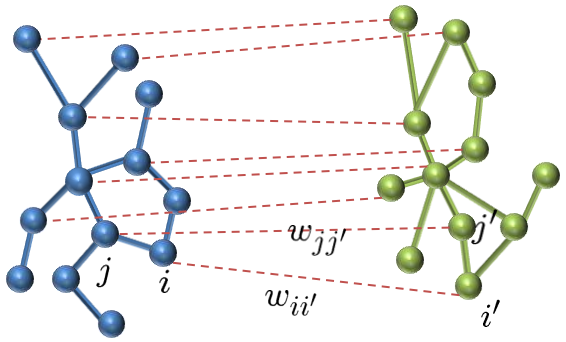
3) Maximum clique.

Quadratic program formulation

maximize $\alpha \sum_{ii'} x_{ii'} w_{ii'} + \beta x^T S x$

Subject to: $\mathbf{A}\vec{x} \preceq \mathbf{1}$

$x_{ii'} \in [0, 1]$



Relaxing the integer constraint→ Still hard (non-concave max.)

Heuristic 1) Find a local maxima using SNOPT → Round to an integer solution.

Naïve linear program (LP) formulation

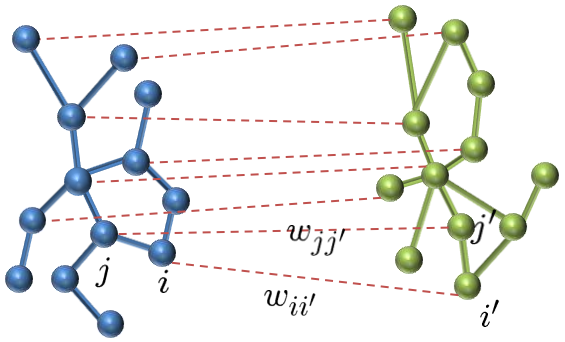
maximize $\alpha \sum_{ii'} x_{ii'} w_{ii'} + \beta \sum_{(ii'jj') \in \mathcal{O}} y_{ii',jj'}$

Subject to:

$$\mathbf{A}\vec{x} \preceq \mathbf{1}$$

$$x_{ii'} \in [0, 1]$$

$$y_{ii',jj'} \leq x_{ii'} \ , \ y_{ii',jj'} \leq x_{jj'}$$



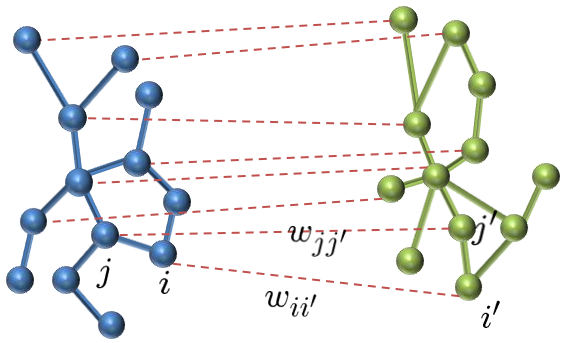
For sparse graphs can be solved relatively efficiently.

Improved LP by Klau'09

maximize $$\alpha \sum_{ii'} x_{ii'} w_{ii'} + \beta \sum_{(ii'jj') \in \mathcal{O}} y_{ii',jj'}$$

$$+ \sum_{(ii'jj') \in \mathcal{O}} u_{ii',jj'} \left(y_{ii',jj'} - y_{jj',ii'} \right)$$

Lagrange multiplier



Subject to:

$$\mathbf{A}\vec{x} \preceq 1$$

$$x_{ii'} \in [0, 1]$$

$$y_{ii',jj'} \leq x_{ii'} \ , \ y_{ii',jj'} \leq x_{jj'}$$

+ some other combinatorial constraints

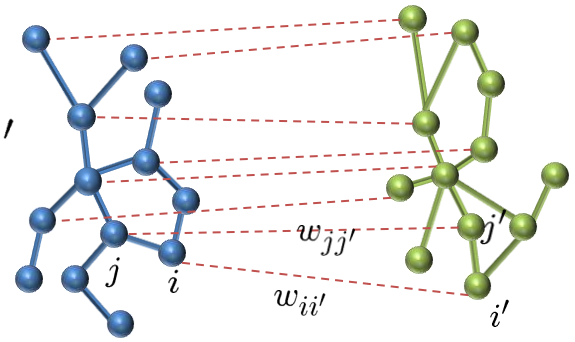
Both LPs and QP also produce an upper-bound for the optimum.

IsoRank (Berger et al'07, 08)

maximize $\alpha \sum_{ii'} x_{ii'} r_{ii'} + \beta \sum_{(ii'jj') \in \mathcal{O}} x_{ii'} x_{jj'}$

Subject to: $\mathbf{A}\vec{x} \preceq \mathbf{1}$

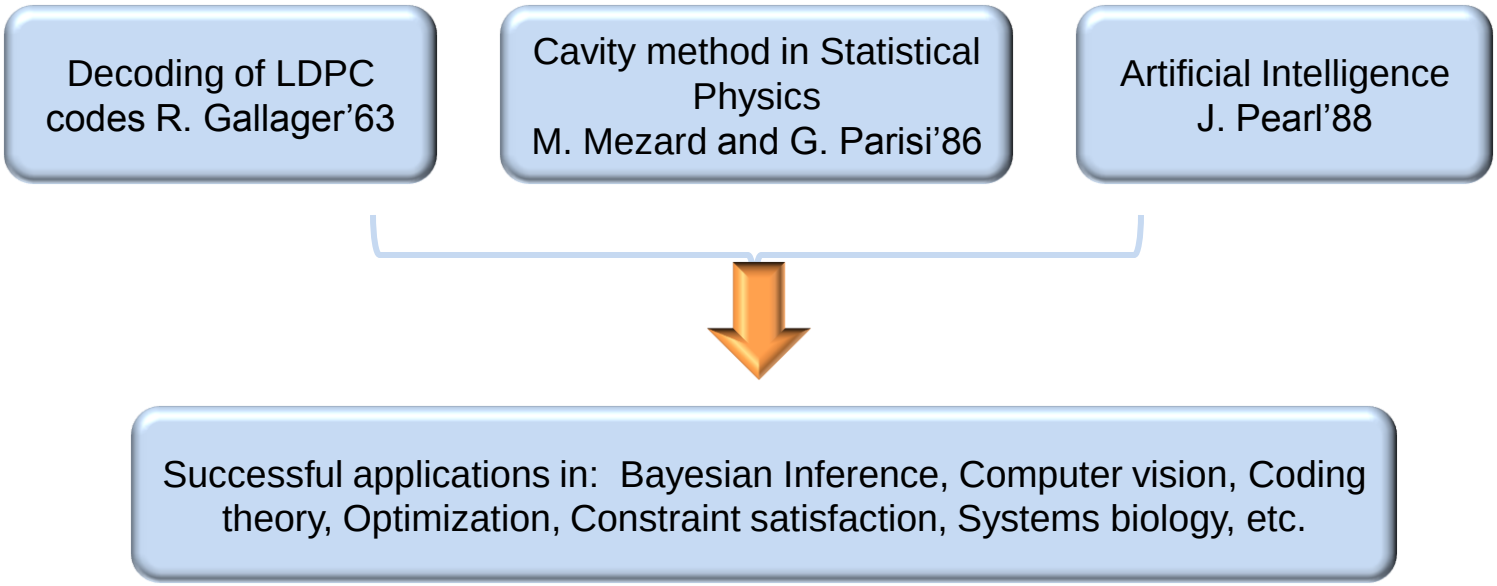
$x_{ii'} \in [0, 1]$



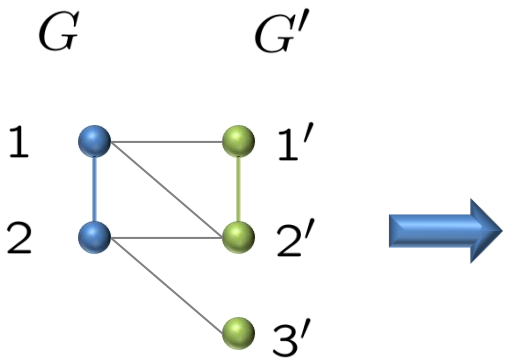
$$r_{ii'} = \sum_{j \in \partial_i} \sum_{j' \in \partial'_i} \frac{r_{ii'}}{|\partial_i||\partial'_i|}$$

The new weights, $r_{ii'}$ ts can be found using an eigen-value calculation (similar to PageRank).

Our approach: Belief Propagation (BP)

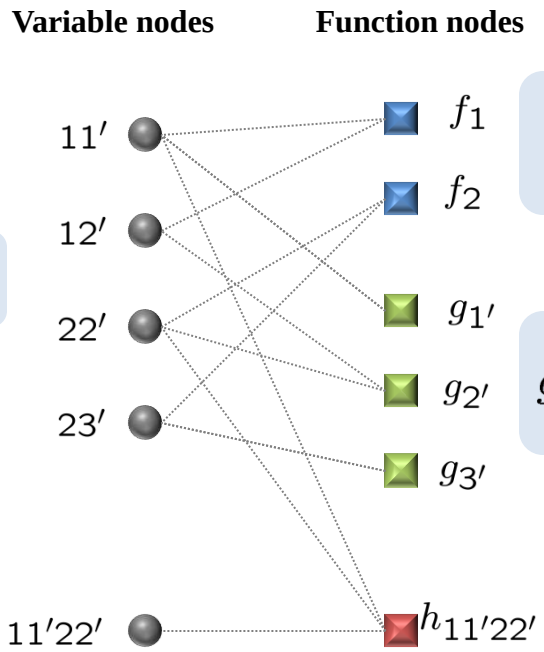


Our approach: Belief Propagation (BP)



$x_{ii'}$

$x_{ii',jj'}$



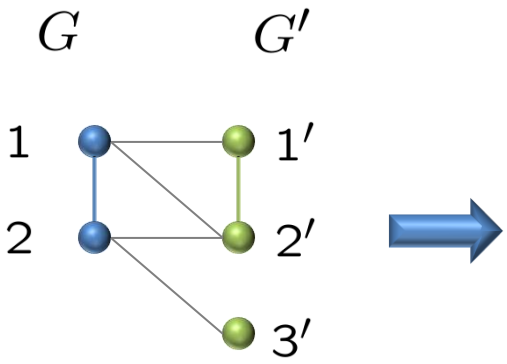
$$f_i([x_{ii'}]_i) = 1_{\{\sum_{i'} x_{ii'} \le 1\}}$$

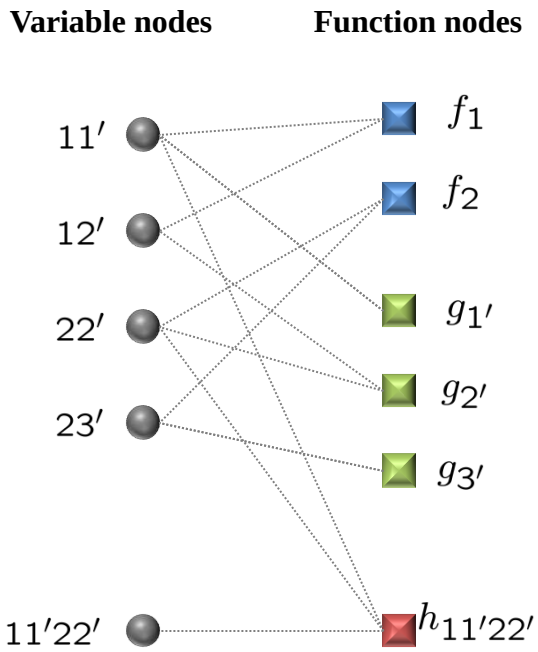
$$g_{i'}([x_{ii'}]_i) = 1_{\{\sum_{i} x_{ii'} \le 1\}}$$

$$h_{ii'jj'}([x_{ii'}, x_{jj'}, x_{ii'jj'}]) = 1_{\{x_{ii'jj'} = x_{ii'} x_{jj'}\}}.$$

Independently, BP was used by Bradde-Braunstein-Mahmoudi-Tira-Weigt-Zecchina'09 for similar problems.

Our approach: Belief Propagation (BP)





$$p(\bar{x}_{E_L}, \bar{x}_S) \propto e^{\alpha \bar{w}^T \bar{x}_{E_L} + \frac{\beta}{2} I^T \bar{x}_S} \prod_{i=1}^{n} f_i(\bar{x}_{\partial f_i}) \times \prod_{i'=1}^{m} g_{i'}(\bar{x}_{\partial g_{i'}}) \prod_{ii'jj' \in V_S} h_{ii'jj'}(\bar{x}_{\partial h_{ii'jj'}}).$$

Belief Propagation for $\beta=0$

1) Iterate the following:

For $k = 0, 1, \ldots$ update the following messages on each link of the network.

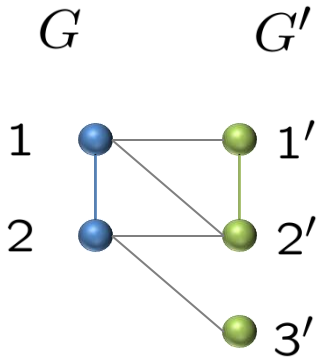
How much i likes to mach to i'

$$m^k_{i \to i'} = w_{ii'} - \max_{j' \neq i'} \left((m^{k-1}_{j' \to i})^+ \right).$$

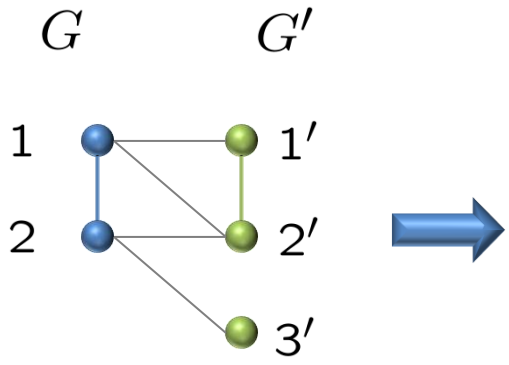
2) The estimated solution at the end of iteration k choose a matching π^k

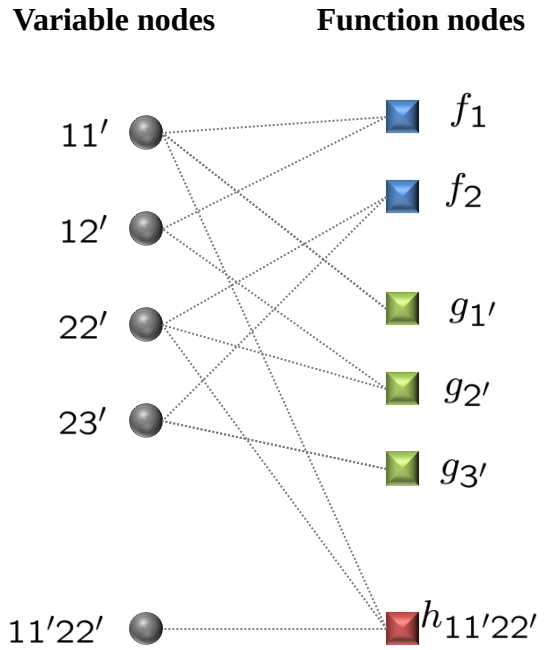
$$\pi^k(i) = \arg \max_{1 \leq j \leq n} \left(m^k_{j' \to i'} \right).$$

i.e. pick the link with maximum incoming message.



Belief Propagation for $\beta>0$





How much *i* likes to mach to *i'*

$$m^{(t)}_{ii'\to f_i} = \alpha w_{ii'} - \left(\max_{k\neq i}\left[m^{(t-1)}_{ki'\to g_{i'}}\right]\right)^+ + \sum_{ii'jj'\in O} \min\left(\frac{\beta}{2}, \max(0, \frac{\beta}{2} + m^{(t-1)}_{jj'\to h_{ii'jj'}})\right).$$

$$m^{(t)}_{ii'\to h_{ii'jj'}} = \alpha w_{ii'} - \left(\max_{k\neq i}\left[m^{(t-1)}_{ki'\to g_{i'}}\right]\right)^+ - \left(\max_{k'\neq i'}\left[m^{(t-1)}_{ik'\to f_i}\right]\right)^+ + \sum_{\substack{kk'\neq jj'\\ ii'kk'\in O}} \min\left(\frac{\beta}{2}, \max(0, m^{(t-1)}_{kk'\to h_{ii'kk'}} + \frac{\beta}{2})\right)$$

How much *ii'* likes to have overlap with *jj'*

Algorithm works for $\beta=0$

(B-Shah-Sharma'05) Each node's decision is correct for $k \geq \lceil \frac{2n \max_{ii'} |w_{ii'}|}{\epsilon} \rceil$

(B-Borgs-Chayes-Zecchina'07): Same algorithm works for any graph when LP relaxation of the problem is integral.

- Generalizes to b-matchings. (independently by Sanghavi-Malioutov-Wilskey'07).
- Works for asynchronous updates as well.

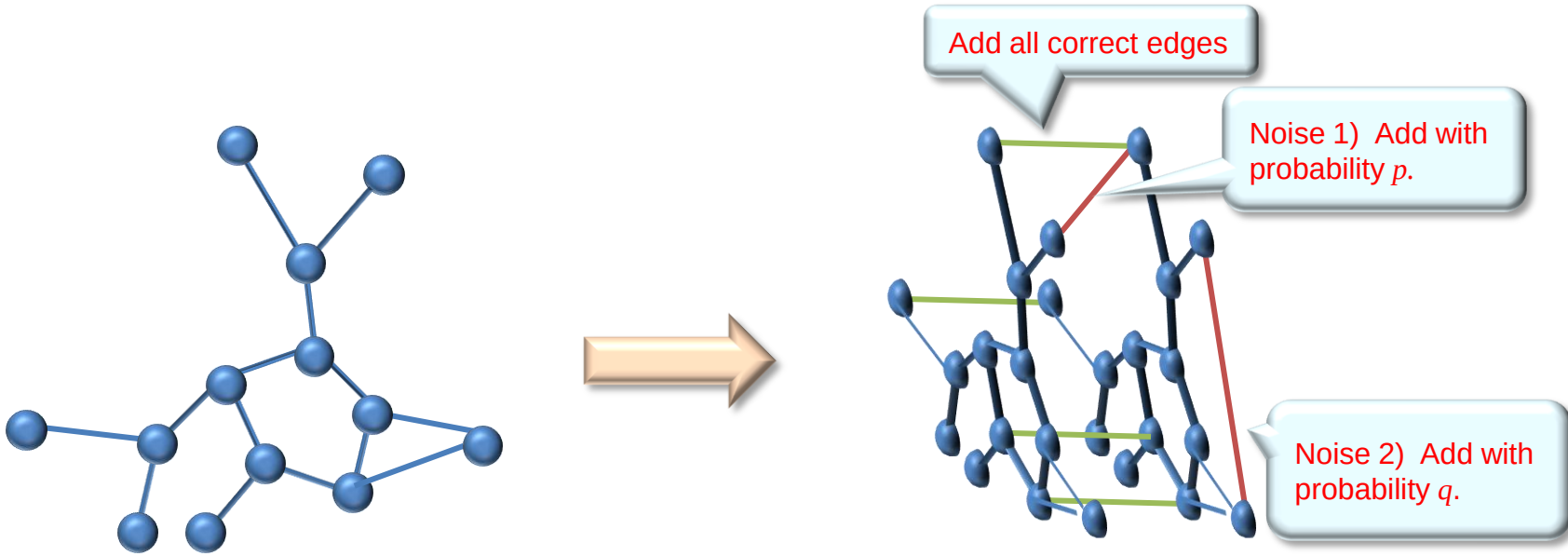
(B-Borgs-Chayes-Zecchina'08): "Belief Propagation" solves the LP relaxation.

- Can use Belief Propagation messages to find the LP solutions in all cases.

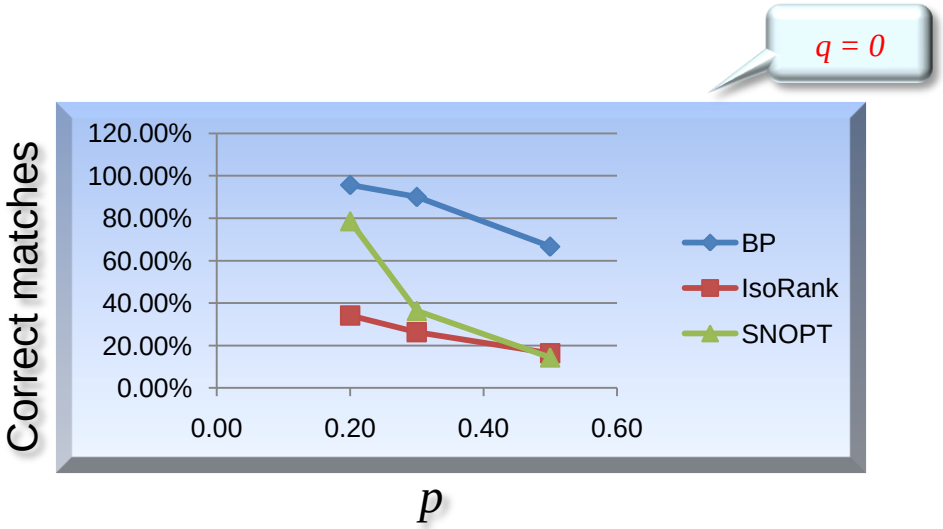
How about the $\beta > 0$?

Experiment on Synthetic data

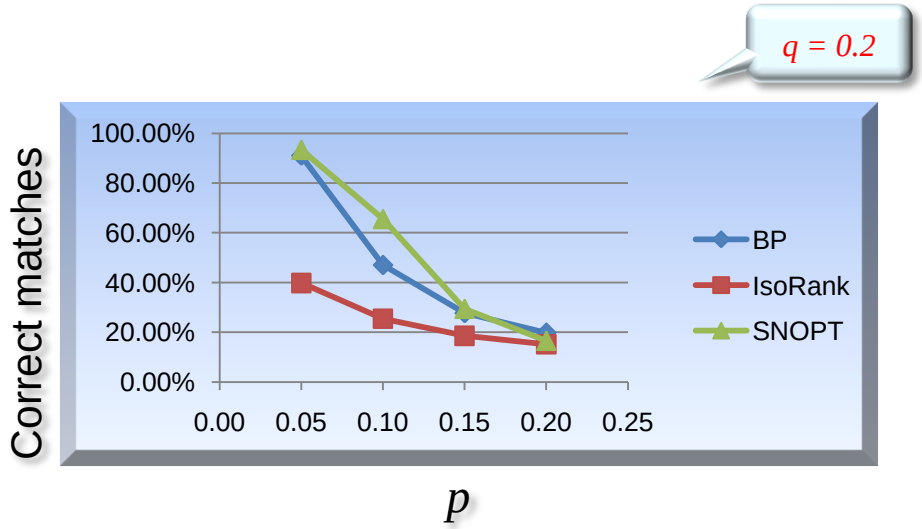
Most of the real-world networks including Wikipedia and LCSH have power-law distribution (The node degree distribution satisfies $P(d_i = k) = \frac{1}{k^\theta}$.)

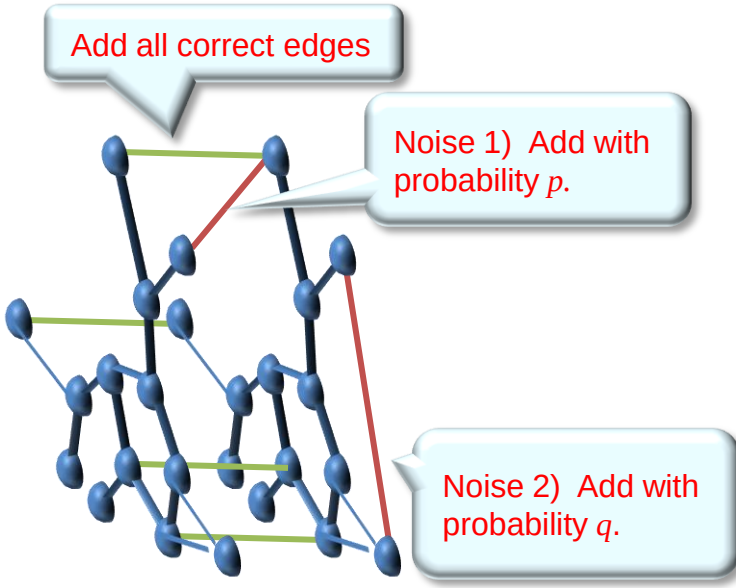


Experiment on Synthetic data

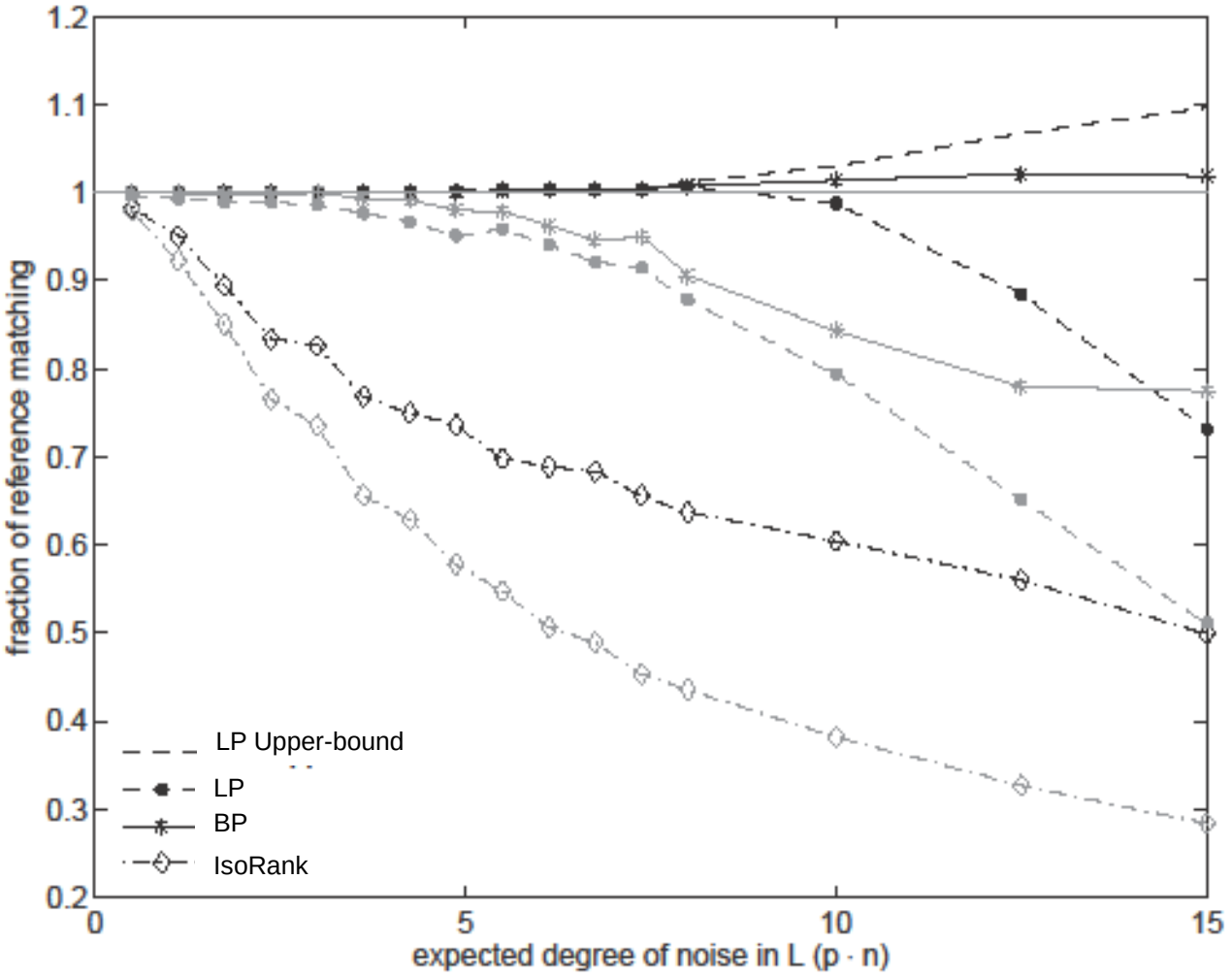


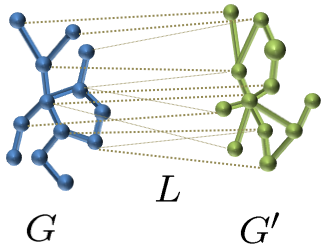
BP, IsoRank → few seconds
SNOPT → few hours



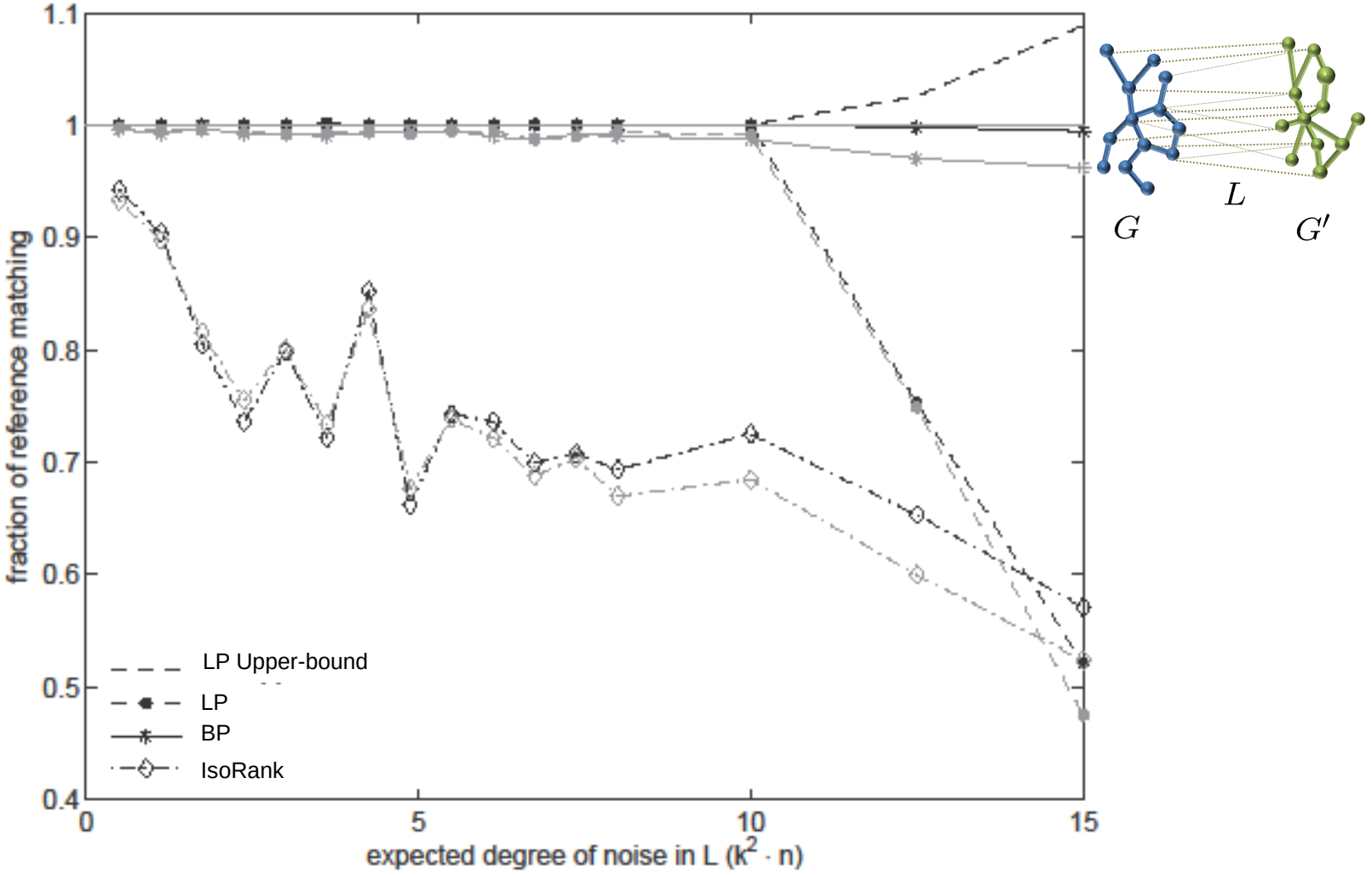


Power-law graph experiments

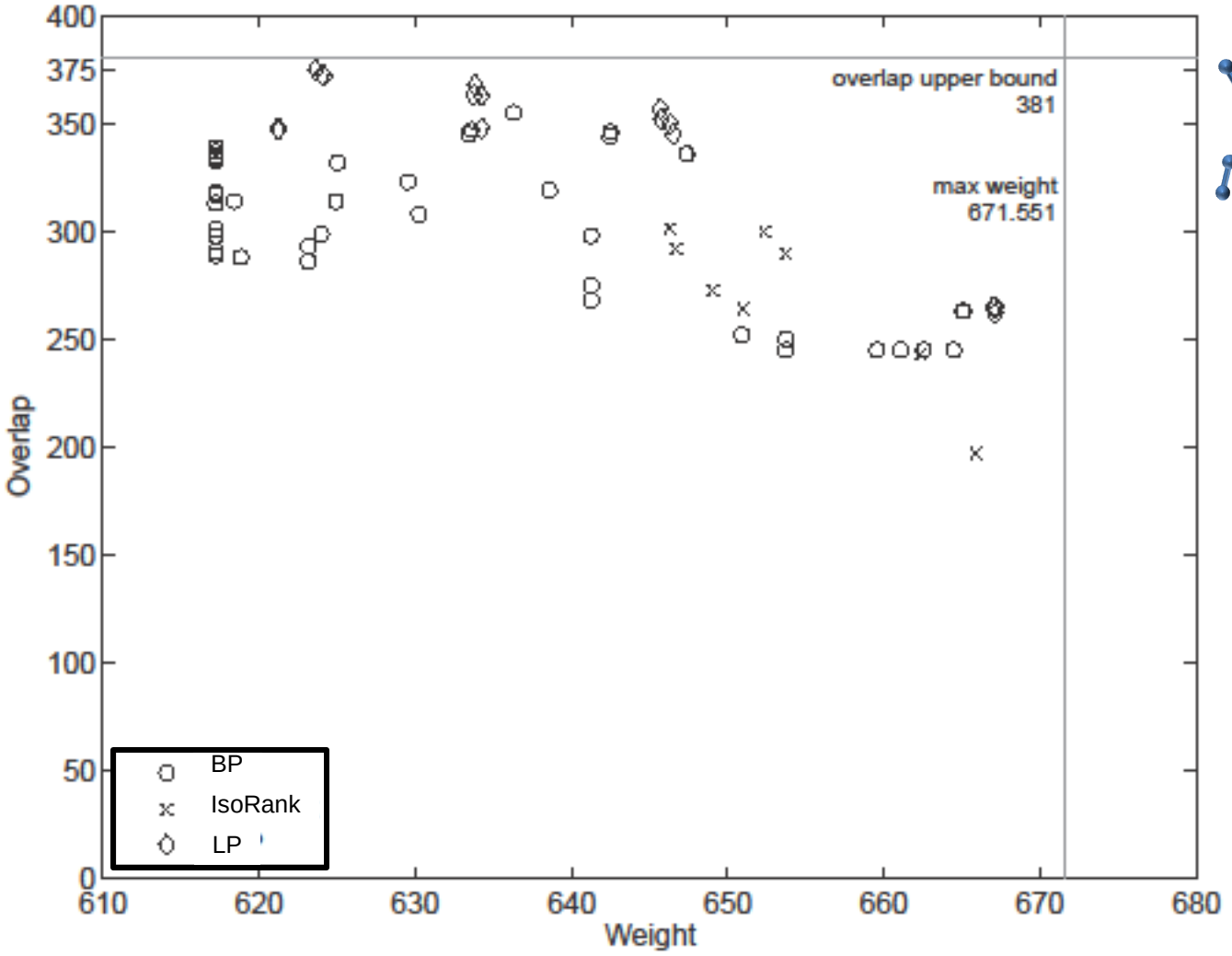


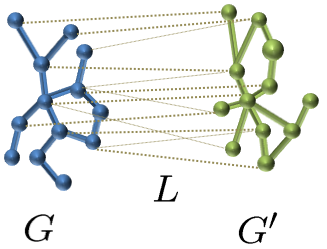


Grid graphs experiments

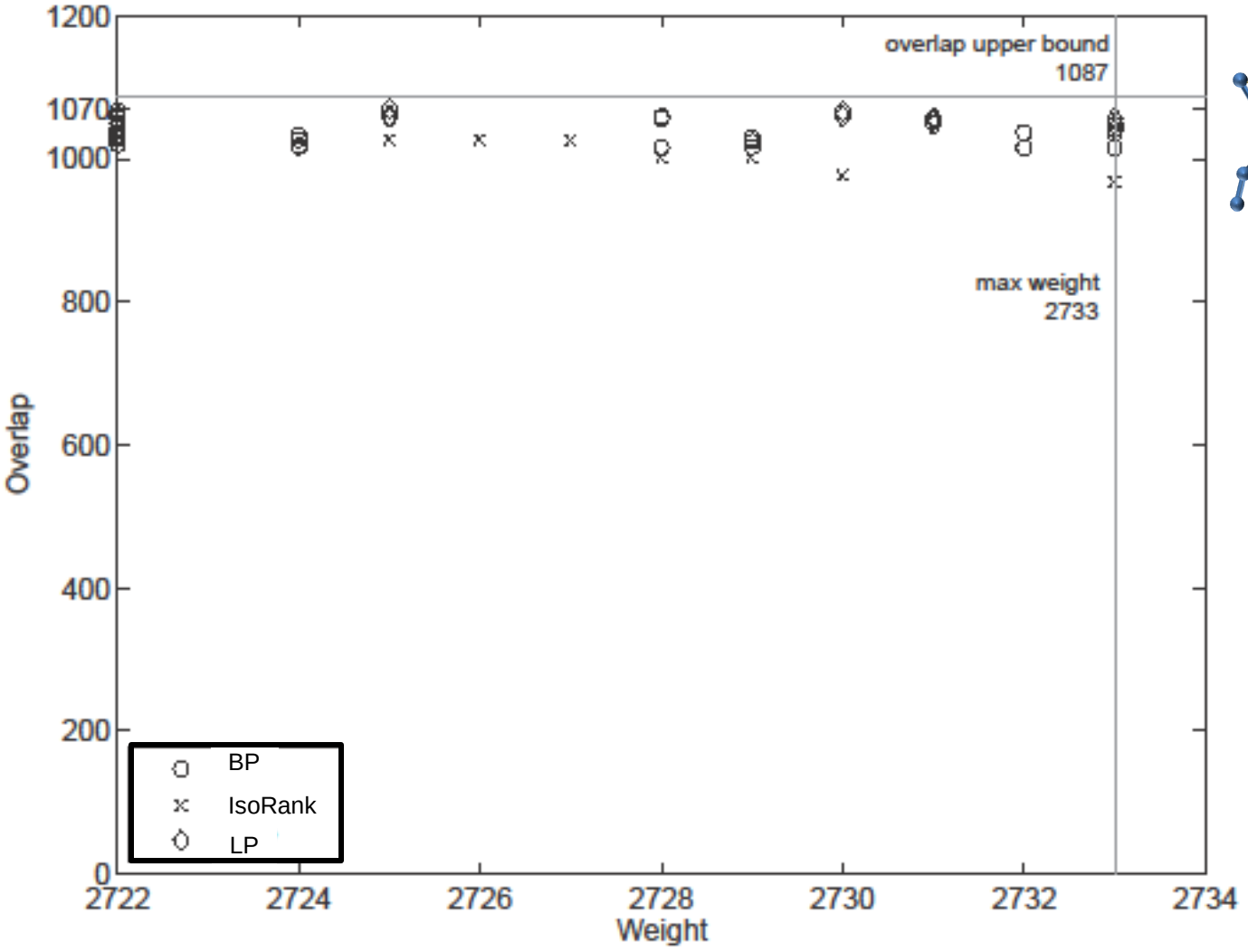


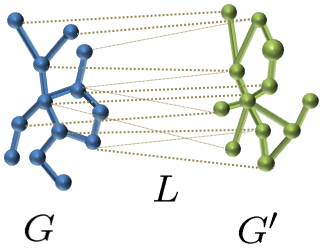
Bioinformatics data: Fly vs Yeast



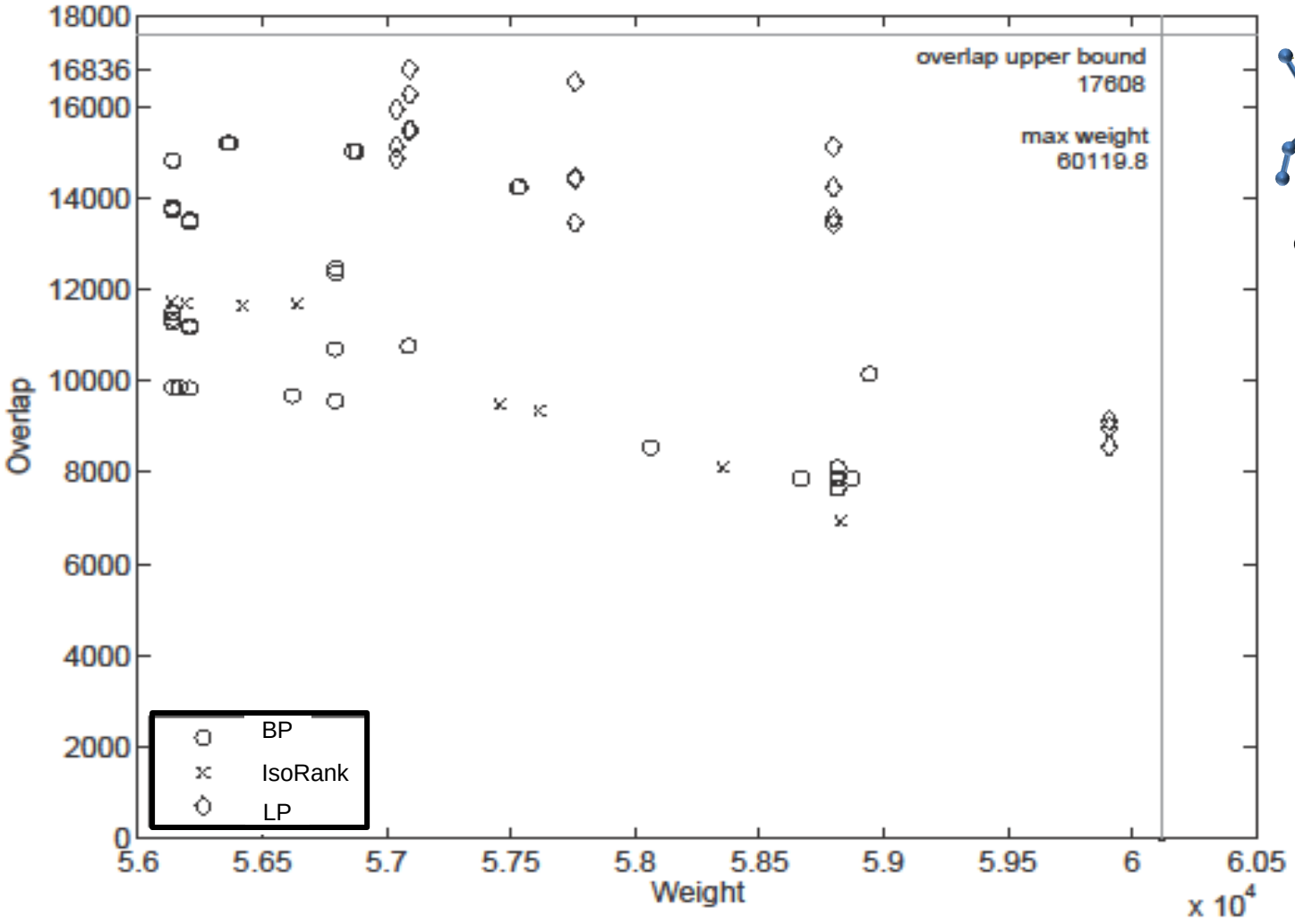


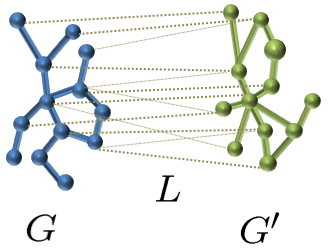
Bioinformatics data: Human vs Mouse





Ontology data: Wiki vs LCSH



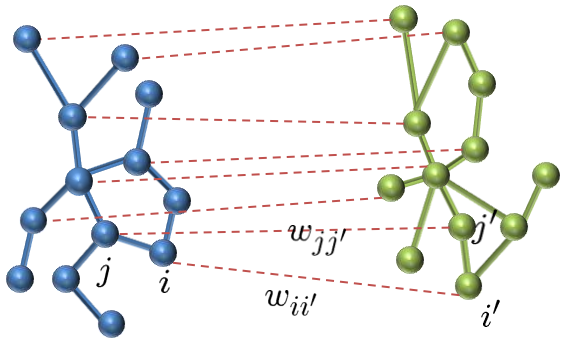


Statistical significance

maximize $\alpha \sum_{ii'} x_{ii'} w_{ii'} + \beta x^T S x$

Subject to: $\mathbf{A}\vec{x} \preceq \mathbf{1}$

$x_{ii'} \in \{0, 1\}$

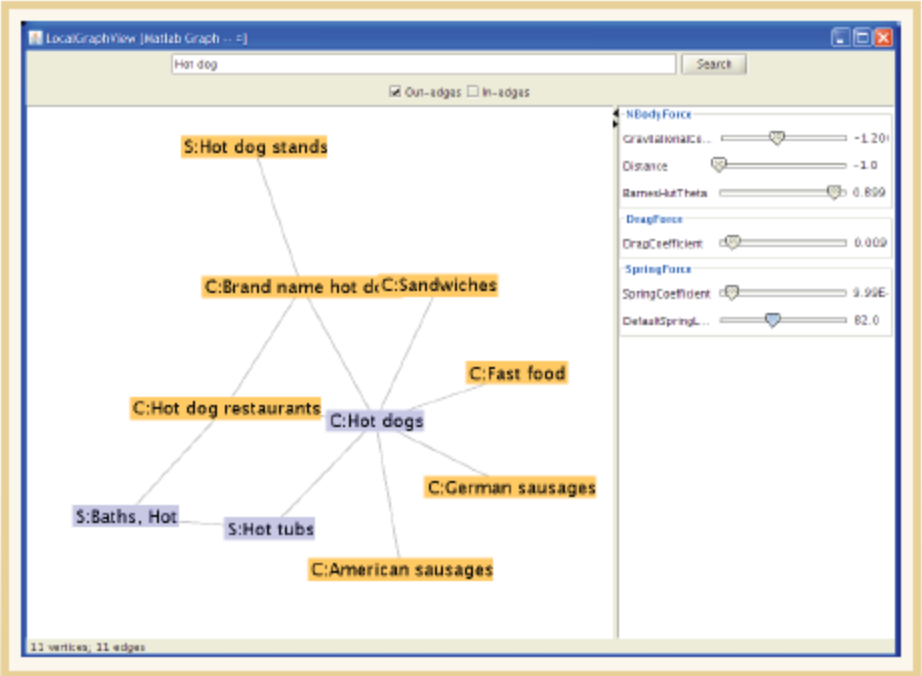


Create many uniform random samples of LCSH and Wiki with the same node degrees. The objective value drops by 99%.

Statistical evidence that the two data-sets are very comparable.

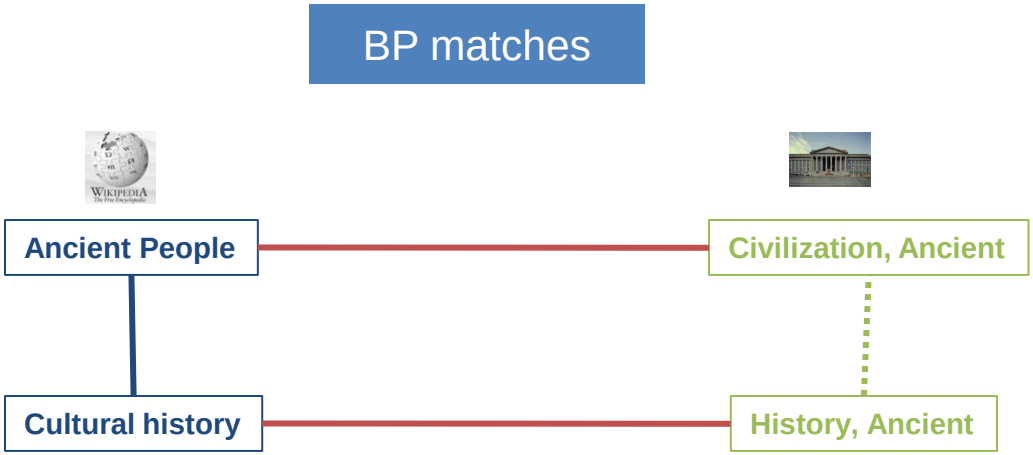
Some matched titles

LCSH	WC
Science fiction television series	Science fiction television programs
Turing test	Turing test
Maching learning	Machine learning
Hot tubs	Hot dog



Enriching the data-sets

- The approach suggests few thousands of potential links to be tested with human experts in the Library of Congress.



Conclusions

- Only BP, IsoRank and LP can handle large graphs.
- BP and LP find near optimum solution on sparse data
- LP produces an upper bound, and slightly better results. But slightly slower.
- For denser graphs BP outperforms LP.

Thank You!